

AFRL-IF-RS-TR-2005-333
Final Technical Report
September 2005



SIMULATION DEVELOPMENT FOR DYNAMIC SITUATION AWARENESS AND PREDICTION

Northrop Grumman Mission Systems

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-333 has been reviewed and is approved for publication

APPROVED: /s/

DAWN A. TREVISANI
Project Engineer

FOR THE DIRECTOR: /s/

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE SEPTEMBER 2005	3. REPORT TYPE AND DATES COVERED Final Jul 04 – Mar 05	
4. TITLE AND SUBTITLE SIMULATION DEVELOPMENT FOR DYNAMIC SITUATION AWARENESS AND PREDICTION			5. FUNDING NUMBERS C - F30602-03-D-0026/0022 PE - 62702F PR - 459S TA - QP WU - 22	
6. AUTHOR(S) Kevin C. Trott				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Northrop Grumman Mission Systems Beeches Technical Campus Rt 26N Rome New York 13441-4505			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFSB 525 Brooks Road Rome New York 13441-4505			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2005-333	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Dawn A. Trevisani/IFSB/(315) 330-7311				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 Words) The Joint Synthetic Battlespace for Research and Development (JSB-RD) is a distributed simulation environment that is intended to facilitate research into both advanced Air Force command and control concepts and simulation science. The JSB-RD environment includes a computer generated forces (CGF) system, the JSB-RD Viewer, which provides theater-level situation awareness and mission status information in multiple formats, weather and environment simulations, a collection of sensor simulations, and a gateway(s) to C4ISR systems. A variety of AFRL/IF programs require a flexible, multi-purpose, modeling and simulation environment in order to represent various aspects of the joint battlespace, and to stimulate and react to prototype C4ISR systems and components. The JSB-RD system forms the starting point for such an environment. This effort focused on Visualization, System Integration and Modeling and Simulation Facility Planning. NG developed improvements and enhancements to the existing JSB-RD Viewer application to allow it to more effectively be used to evaluate and demonstrate advanced visualization and GUI techniques for theater-level situation awareness, integrated the enhanced JSB-RD Viewer application with the other major components of the JSB-RD federation, developed test and demonstration scenarios, and assessed how the existing JSB-RD system, and its individual components, could be used as the starting point for the development of a more complete C4ISR modeling and simulation environment.				
14. SUBJECT TERMS Model Abstraction, Simulation Abstraction, C4ISR, ISR, Modeling and Simulation, JSAF			15. NUMBER OF PAGES 30	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

1. Introduction.....	1
2. JSB-RD Distributed Simulation Environment.....	2
2.1 Overall Architecture	2
2.2 JSB-RD Components	3
2.2.1 JSAF	4
2.2.2 OASES.....	6
2.2.3 DT-SIM.....	9
2.2.4 Clutter Simulation	9
2.2.5 MC02 Federation Object Model.....	10
2.2.6 Integrated Situation Viewer	10
2.2.7 Simulation Preparation	17
2.2.8 GIEsim	17
3. Lessons Learned	18
3.1 JSAF Modification.....	18
3.2 JSAF Scenario Preparation.....	20
3.3 HLA-to-XML Gateway	20
3.4 Integrated Situation Viewer	21
3.5 JSAF-GIEsim Integration.....	21
3.6 Conclusions and Recommendations.....	21
4. Acronyms	24

LIST OF FIGURES

Figure 2.1	JSB-RD Overall Architecture	2
Figure 2.2	JSB-RD Components.....	3
Figure 2.3	JSAF Plan View Display with Air Dynamics Editor	6
Figure 2.4	OASES Organization	8
Figure 2.5	OASES Weather Visualization.....	9
Figure 2.6	Viewer Architecture.....	11
Figure 2.7	Main (Map View) GUI.....	12
Figure 2.8	Lens View.....	13
Figure 2.9	Magnified Map View.....	14
Figure 2.10	Table View	15
Figure 2.11	SNN View.....	16

1. INTRODUCTION

This final technical report summarizes the accomplishments and lessons learned under the Simulation Development for Dynamic Situation Awareness and Prediction (Sim Dev for DSAP) contract. This work was performed by Northrop Grumman Mission Systems (NGMS) for the AFRL, C4ISR Modeling and Simulation Branch (IFSB), under Contract F30602-03-D-0026/0022. This report addresses CLIN 0002, CDRL A007 of that contract.

The overall objective of this effort was to develop a flexible, closed-loop distributed simulation environment, in which detailed mission plans can be developed, be used as input to a set of distributed simulations, and be executed within the simulation environment. These simulations provide feedback to prototype Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems in the form of mission status reports, sensor tracks, and other ISR mission results reports, which can be used to maintain situation awareness and to dynamically adjust mission plans in response to events. This work was carried out within the context of the Joint Synthetic Battlespace for Research and Development (JSB-RD), a distributed C4ISR modeling and simulation environment at the AFRL Rome Research Site. The JSB-RD synthetic battlespace is designed to address three objectives:

1. To provide a testbed for C4ISR research, experimentation, and evaluation, supporting AFRL C4ISR concepts and programs such as Predictive Battlespace Awareness (PBA), the Commander's Predictive Environment (CPE), Effects Based Operations (EBO), and using communications infrastructures being developed by AFRL, such as the Joint Battlespace Infosphere (JBI).
2. To explore the synchronization of simulations to real-world situation data in order to predict future events, supporting concepts such as Dynamic Situation Awareness and Prediction (DSAP) and Course of Action (COA) Analysis.
3. To support research into simulation science, particularly in the areas of adversarial modeling, information management, and visualization.

The JSB-RD distributed simulation environment was constructed primarily by integrating existing simulations and tools. The JSB-RD environment is centered around the Joint Semi-Automated Forces (JSAF) simulation software. JSAF is a computer generated forces (CGF) system that is used by the U.S. Joint Forces Command for joint experimentation. The JSB-RD environment also includes the Ocean, Atmosphere, and Space Environmental Services (OASES) system, which models weather, and the Dynamic Terrain Simulation (DT-SIM), which models changes to the environment such as bomb craters, damage to buildings, and the creation and destruction of obstacles, as well as a clutter simulation, which models civilian traffic. The environment also includes tools for creating scenarios from existing Air Battle Plans, extracted from the Air Operations Data Base (AODB) with the Theater Battle Management Core System (TBMCS), as well as gateways for connecting simulations communicating using DMSO's High Level Architecture (HLA) and/or the earlier Distributed Interactive Simulation (DIS) protocol, with C4ISR systems, using AFRL's JBI. Under this effort, the Global Information Enterprise

Simulation (GIEsim) communication simulation was added to the JSB-RD environment to provide Link-16 network modeling.

Section 2 describes the JSB-RD environment in more detail. Section 3 summarizes the lessons learned in the course of performing this effort, including recommendations for subsequent related work. Section 4 contains a list of acronyms.

2. JSB-RD DISTRIBUTED SIMULATION ENVIRONMENT

As noted above, the JSB-RD distributed simulation environment consists of a number of components, which are connected together using several different mechanisms. Fundamentally, the environment is an HLA federation made up of simulations that communicate using a variation of the Millenium Challenge 2002 (MC02) Federation Object Model (FOM). Simulations that still use the DIS protocol can be connected through a DIS-HLA gateway federate. Similarly, an HLA-JBI gateway allows the federation to communicate with C4ISR system component prototypes being developed at AFRL.

2.1 Overall Architecture

The overall architecture of the JSB-RD distributed simulation environment is shown in Figure 2.1. It consists of three primary components, addressing each of the three primary aspects of any simulation environment.

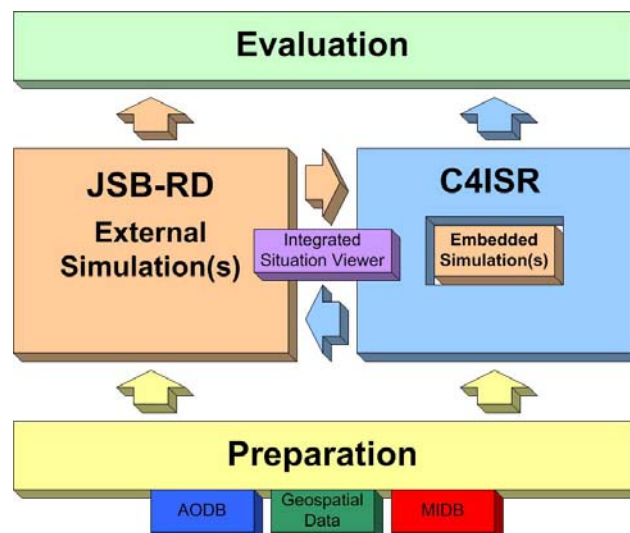


Figure 2.1: JSB-RD Overall Architecture

At the bottom, the Preparation component addresses experiment planning, scenario preparation, and all other aspects of preparing a simulation experiment. This includes accessing, extracting, and manipulating various kinds of source data, including Air Battle Plan (ABP) and Friendly Air Order of Battle information contained in the AODB, and Enemy Order of Battle (EOB) information contained in the MIDB, as well as geospatial data. In the middle are the components that address the execution phase of a simulation experiment. These consist of the various

simulations that make up the JSB-RD environment, as well as real C4ISR systems, system components, or prototypes. Note that the C4ISR component can also include an embedded simulation within it, used for COA analysis or other forms of prediction. The simulation and C4ISR components communicate with one another in both directions. The status of friendly entities and of visible opposing entities, and the effects of any relevant scenario events, are reported by the simulation to the C4ISR system, via various modeled sensor and communication systems. As the C4ISR system issues commands, they are passed back to the simulated entities that are to carry them out. In the center, visualizing both the simulation and C4ISR information, either separately or together, is the Integrated Situation Viewer application.

At the top, the Evaluation component provides a collection of tools for analyzing data produced by both the simulation and C4ISR systems. This component addresses the post-execution aspects of a simulation experiment. It is currently the least developed part of the JSB-RD environment.

2.2 JSB-RD Components

The components of the JSB-RD distributed simulation environment are shown in Figure 2.2.

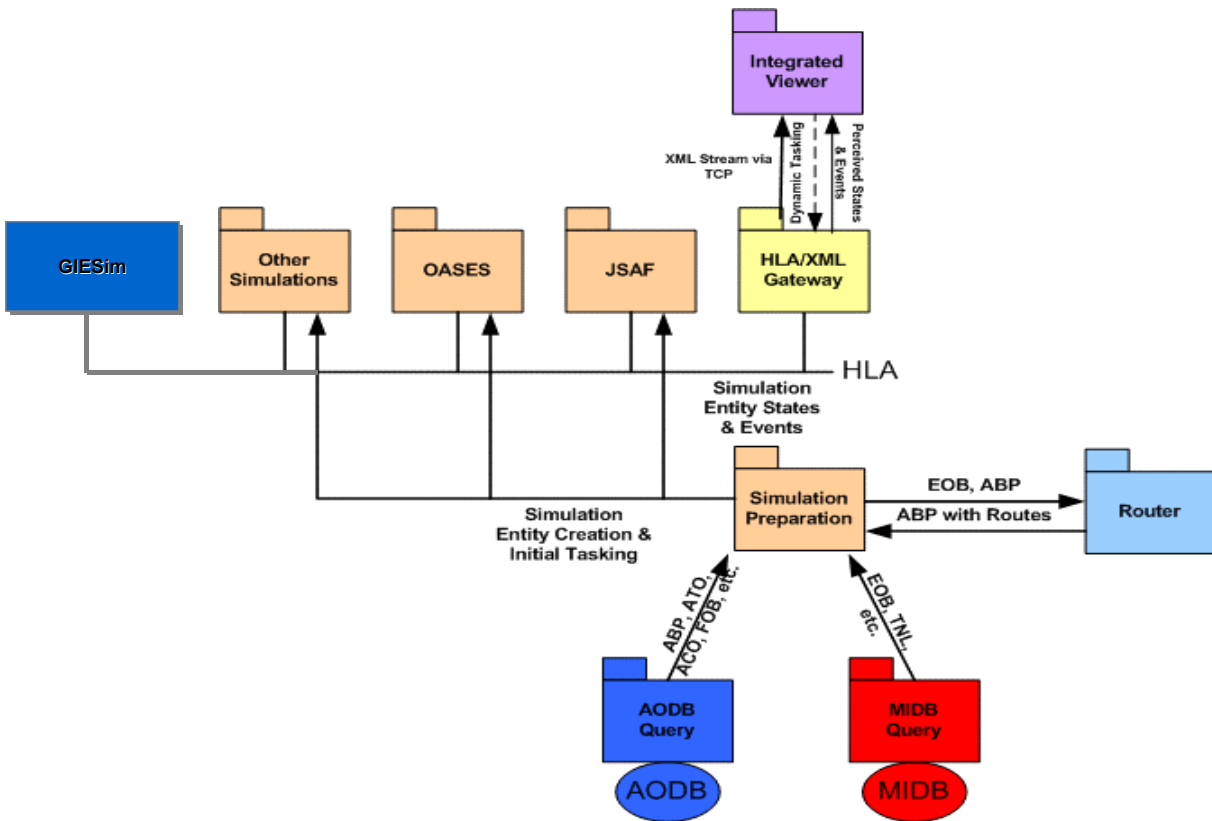


Figure 2.2: JSB-RD Components

JSAF, OASES, GIESim, and several supporting simulations make up an HLA federation. The HLA/XML Gateway translates the federation message traffic into an XML stream that can be easily read by a variety of applications. One such application is the Integrated Situation Viewer, which receives and displays entity state and interaction information. The Simulation Preparation

tool extracts Air Battle Plan and Friendly Order of Battle information from the AODB, and Enemy Order of Battle information from the MIDB, which it then uses to generate a scenario input spreadsheet that can be read by JSAF to create and task the necessary simulation entities. The Simulation Preparation tool also interfaces with an aircraft route-planning tool created at AFRL to generate more realistic air mission flight paths that avoid known threats.

2.2.1 JSAF

JSAF is currently the primary simulation used within the JSB-RD environment. As noted above, JSAF is an entity-level, computer generated forces (CGF) simulation system that is used by the U.S. Joint Forces Command for joint experimentation, by the U.S. Navy for Fleet Battle Experiments, and by the AFRL Human Effectiveness Directorate (AFRL/HE) in support of the Distributed Mission Training (DMT) program. JSAF provides entity-level simulation of ground, air, and naval forces. In support of the joint experiment Urban Resolve, it has been used by JFCOM to simulate more than 100,000 entities within a single distributed simulation. It has also been used to support a variety of experiments with environment simulation, including dynamic terrain (i.e., craters, trenches, etc.), weather, and chemical/biological warfare defense, as part of DMSO's EnviroFed program. JSAF was originally developed by DARPA as part of its Synthetic Theater of War (STOW) program, and is descended from ModSAF. JSAF is maintained by the Joint Forces Command (JFCOM). It incorporates intelligent agents, due to the Soar program, that autonomously controls simulated fixed wing aircraft.

JSAF is an extremely large software application. It was originally developed more than 15 years ago, in C, and has been extensively modified and extended. As a result, its original architecture has been almost completely obscured. It now consists of more than 1000 object libraries that model different types of platforms, weapons, sensors, etc. While it contains a great deal of powerful simulation functionality, it is difficult to use, and even more difficult to modify. Documentation and support are both extremely limited.

JSAF runs under the Linux operating system. In the JSB-RD environment, JSAF can be run on multiple Linux systems simultaneously. The individual copies function together as a single HLA federate, keeping their separate internal database copies in synchronization, and sharing the computational load.

The primary input to JSAF is a spreadsheet file that defines a collection of entities, both friendly and enemy, to be created, and specifies how each entity is to be tasked. Entity types, initial locations, call signs, and assigned tasks are identified. Such spreadsheets can be prepared by hand. However, they can also be automatically generated from Air Battle Plan information, supported by Friendly and Enemy Order of Battle information, extracted from the AODB and MIDB within TBMCS.

JSAF scenarios can also be created interactively, using its integral Unit Editor. Individual entities and small units can be created, placed on the Plan View Display, and assigned tasks to perform. Interactively created scenarios can be saved and (re)loaded. However, such scenarios and the scenario spreadsheets are two separate mechanisms, and cannot be readily combined. There is currently no mechanism that allows JSAF entities to be tasked by other federates.

JSAF is capable of receiving and processing several types of information via HLA. This includes entity state information output by other simulations within the federation. For example, JSAF reads the states of civilian vehicles and pedestrians that are modeled by the clutter simulation, and displays them on the JSAF Plan View Display. Any such external entities are visible to the JSAF-controlled entities; they can be detected, fired at, collided with, etc. JSAF also reads the weather state information output by OASES although most platform and equipment models within JSAF do not make use of such information. JSAF also reads messages describing changes to the terrain that are output by DT-SIM. Such dynamic terrain changes (e.g., the appearance of a bomb crater) can affect the movement of ground vehicles in JSAF.

The primary output of JSAF is entity state information for all of the entities that it models. It also outputs various types of interactions, including weapons fire and detonation, collision, etc. In support of various exercises and experiments, JSAF has been extensively modified to support additional FOMs. It includes an extensive FOM agility layer. Many of its outputs are platform- or weapon-system specific.

Figure 2.3 shows the JSAF Plan View Display (PVD), with a map background (in this case showing terrain elevation and roads) and various platform-level icons. The toolbar at the upper right provides access to a variety of tools for creating, tasking, and otherwise manipulating the simulated entities. The window at the bottom shows the Aircraft Dynamic control panel, which can be used to manipulate the flight and fuel consumption characteristics of a particular aircraft. Multiple copies of the JSAF PVD can be run simultaneously. Commonly, some JSAF GUIs are configured as controller workstations, which see and can manipulate all entities, while others are configured as “player” workstations, which can see only their own forces, as well as any enemy forces that have been detected.

Under this effort, JSAF was modified to interface it with the GIESim communications simulation software. Several additional interactions were added to the MC02 FOM. Because GIESim only processes HLA interactions, not HLA object messages, all JSAF-GIESim messages are cast as interactions. In particular, the GIESIM_ENTITY_STATE is defined as an interaction. This message contains entity state information in a form tailored for GIESim, including an entity id, entity type information, the location of the entity in geodetic coordinates (latitude and longitude in decimal degrees, altitude in meters), and heading. GIESim uses this information to keep track of the locations of all relevant JSAF entities.

The GIESIM_MSG_SEND and GIESIM_MSG_RCVD interactions provide the mechanism by which GIESim is used to evaluate potential communications between JSAF entities. The GIESIM_MSG_SEND interaction is sent from JSAF to GIESim whenever one JSAF entity desires to communicate with another. It identifies the originator and desired receiver of the message, as well as a unique message identifier, the message length, and the GIESim network type to be used, which is a function of the type of message to be sent. The GIESIM_MSG_RCVD interaction is returned from GIESim to JSAF. It identifies the receiving entity, the message identifier, the GIESim network identifier, and the cumulative latency, in seconds, before the message is actually received.

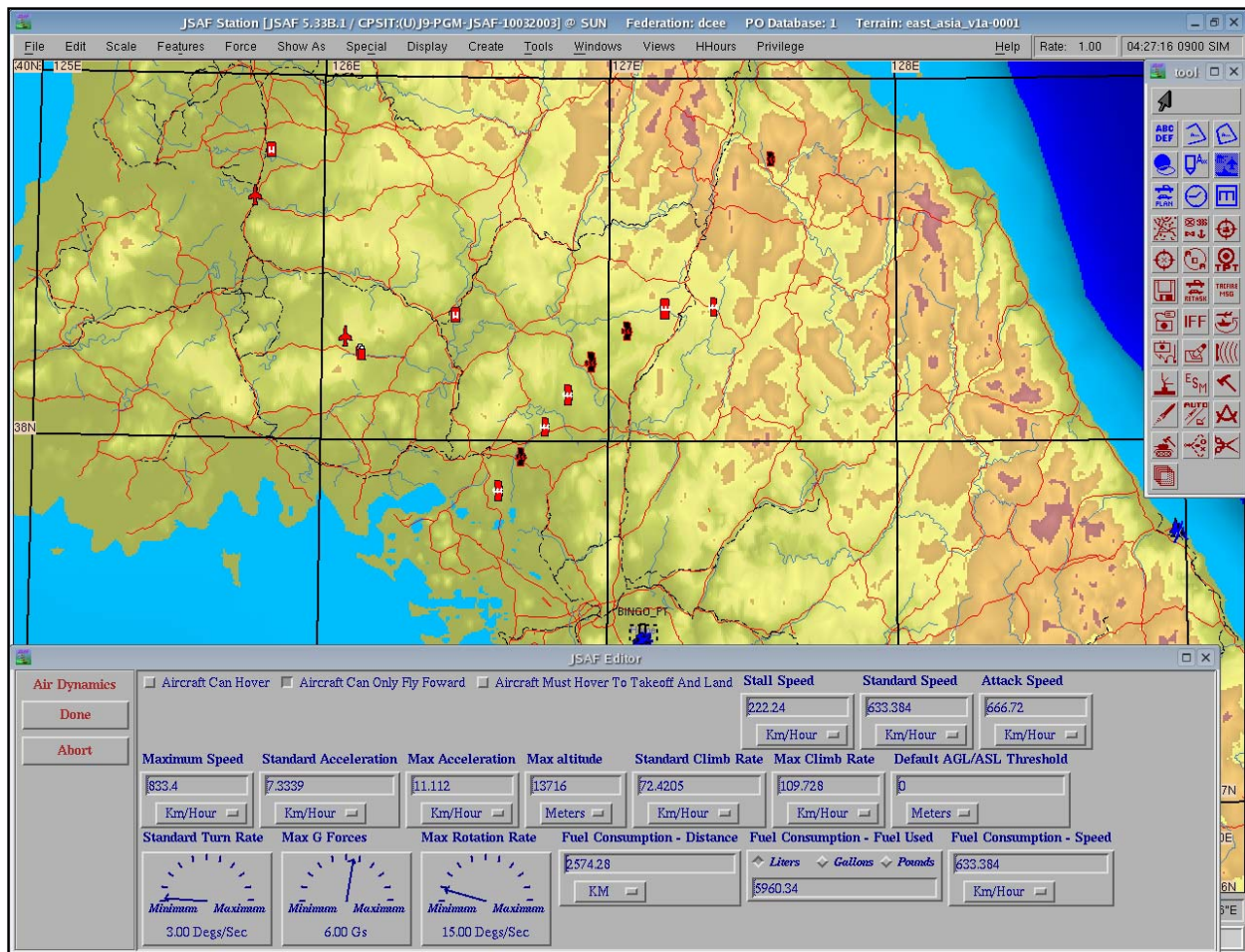


Figure 2.3: JSF Plan View Display with Air Dynamics Editor

JSF software was modified to send the GIESIM_ENTITY_STATE and GIESIM_MSG_SEND interactions, and to receive the GIESIM_MSG_RCVD interaction. JSF reads a script of communications actions, including sender, receiver, message type, and time, from a file, and maintains a queue of GIESIM_MSG_SEND interactions that have been output, but for which a response has not yet been received. When a response is received, in the form of a GIESIM_MSG_RCVD interaction, indicating that the communication is successful, and indicating the communication latency, the message is moved to a second queue until the latency time has elapsed. At that time, an action on the part of the receiving entity, such as another communication, may be triggered.

2.2.2 OASES

The OASES system is a suite of applications for creating and managing a three-dimensional, time-varying, digital representation of the natural environment. OASES has been used primarily to provide synthetic natural environments (SNEs) to systems of networked military training simulations running on the Department of Defense's (DoD) High Level Architecture (HLA). The simulated natural environments created by OASES are based on authoritative, validated

numerical models, typically the same models that are used by METeorological/Oceanographic (METOC) personnel in support of real-world military operations. OASES provides tools for converting authoritative model outputs to a data format recognized by all of the OASES applications. This format supports the data access requirements of distributed simulations that integrate virtual and/or live entities and which must operate in real-time. Additionally, OASES provides tools for tailoring the SNE, either before the simulation begins or while it is running, to meet exercise-specific requirements for environmental phenomena.

The original development of the OASES system was sponsored by the Defense Advanced Research Project Agency (DARPA), under the name *TAOS* (Total Atmosphere Ocean Services), in support of the Synthetic Theater of War (STOW) 1997 Program. In 1998, DARPA funded the development of a low-resolution worldwide atmospheric and oceanographic database, also known as the *Global-98* database, for use by the JSIMS program. In 1999, the United States Space Command's (USSPACECOM) Space Warfare Center funded extensions to TAOS to support the space environment, specifically ionospheric effects on precision-guided missiles, as part of the PSM+ (extended Portable Space Model) project. More recently, funding for continued development and integration with the HLA, under the name OASES, has been provided primarily by DMSO through the Environment Federation (EnviroFed) projects.

Figure 2.4 shows the OASES subsystems and the data flows among them. OASES converts current and forecast environmental data, provided by physics-based operational and research models, into a form that can be used by distributed simulations running on the HLA.

The OASES system consists of five primary subsystems. The OASES Ingestor converts all input model data to a common run-time format that is recognized by all OASES subsystems. The external data providers are identified in the rounded-rectangles on the left side of the figure. The OASES Transformer uses a set of transformation algorithms to augment existing OASES databases with various environmental parameters that are not provided directly by an external data source, but that are required by the simulations served by OASES. The OASES Editor allows users to tailor the contents of an OASES database. The Editor provides three editing algorithms: 1) replacement at a point with gaussian spatial and temporal blending, 2) a Pressure Field Modification (PFM) algorithm for editing atmospheric environments while preserving correlation between temperature, pressure, wind and relative humidity, and 3) a precipitation editing algorithm. The Editor can be used to prepare scripted changes to an existing METOC scenario, or it can be used at run-time to modify the SNE during a simulation exercise. The OASES Time Federate and Publisher are the subsystems that interface directly to the HLA Federation; that is, they are the OASES *Federates*. They create and update the objects that encapsulate the state of the simulated natural environment via services provided by the RTI. The Time Federate creates objects that establish the time-dependence of the SNE while the Publisher manages objects that encapsulate its spatial-dependence.

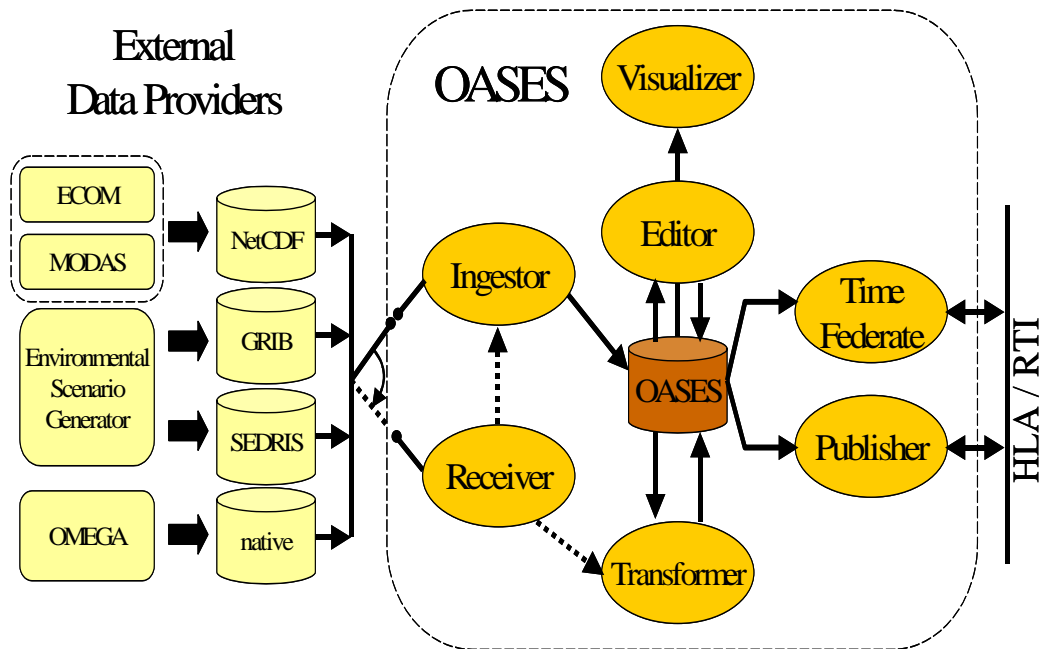


Figure 2.4: OASES Organization

The OASES Visualizer is a tool for visualizing the contents of an OASES database. It is used to validate databases built by the Ingestor and/or extended by the Transformer, to review the results of edits applied by the OASES Editor, to monitor the current state of the SNE created by the Publisher, and/or to monitor the current state of the SNE as received by the OASES Subscriber. An example of an OASES Visualizer display is shown in Figure 2.5.

Finally, the OASES Receiver is responsible for polling local or remote data sources, using the Internet File Transfer Protocol (FTP), for environmental data transmittals matching a user-specified file-naming pattern. The Receiver is the subsystem that supports the “Live Mode” of OASES, in which the simulated natural environment is continuously updated based on the data received from current and forecast environmental models running in real-time.

Within the JSB-RD environment, OASES is used to bring in weather information from Air Force and Navy sources. OASES outputs weather state information, including temperature, pressure, and precipitation information, in one-dimensional (profile), two-dimensional (surface), and three-dimensional forms, over HLA. This information is read by JSAF and DT-SIM, which use it to modify some military operations, and to implement changes to the terrain database, respectively.

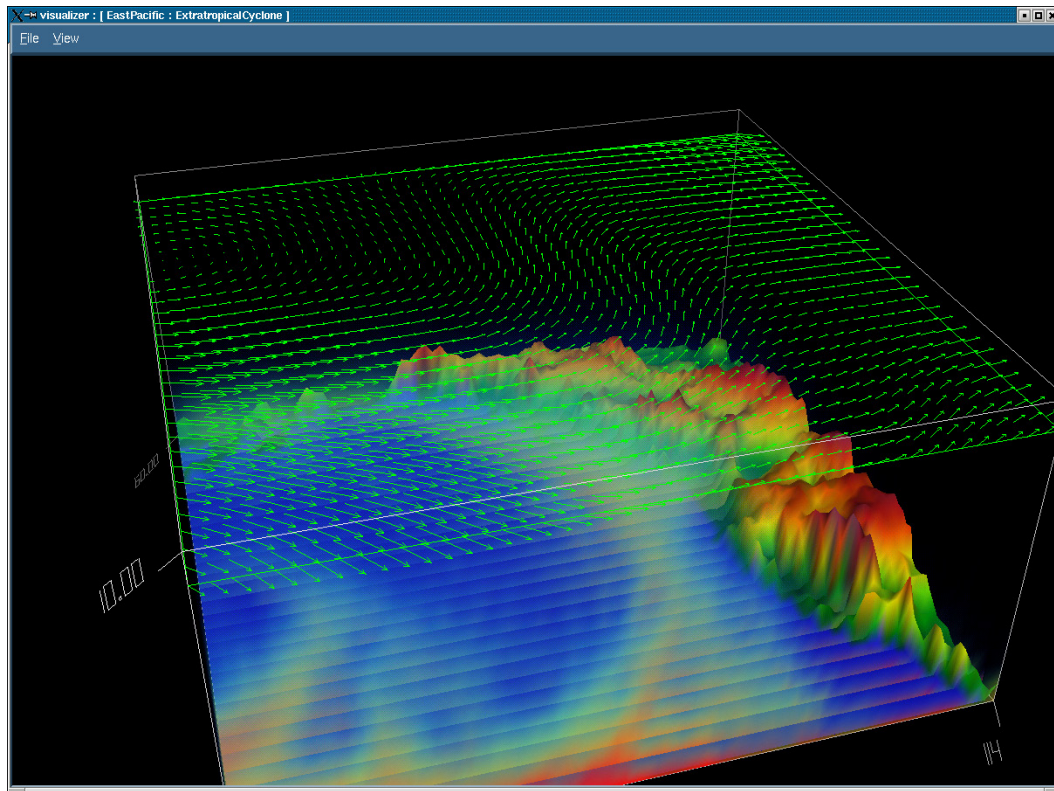


Figure 2.5: OASES Weather Visualization

2.2.3 DT-SIM

The Dynamic Terrain Simulation (DT-SIM) models changes to the terrain component of the environment. The changes result from various types of simulation events, including weapon detonations, movement, weather effects, and military engineering operations, such as the creation and destruction obstacles. DT-SIM receives interaction events from JSAF, and determines what effect, if any, they have on the geometry or attributes of the terrain at the location event. For example, when DT-SIM receives a weapon detonation interaction from JSAF, it may, depending on the type of the weapon and its proximity to the terrain surface, or a specific terrain feature, determine that a crater should be created, or that a building should be damaged. DT-SIM updates its internal terrain representation, and publishes messages over HLA describing how the terrain has changed. These messages are used by JSAF to update its terrain representation, which in turn affects how some activities are carried out. For example, the appearance of a new crater may change the movement of vehicles to avoid it. Weather effects, such as prolonged rain, may also change the characteristics of the terrain, restricting the speed of cross-country movement.

2.2.4 Clutter Simulation

The clutter simulation, which is part of the JSAF distribution, models the movements of civilian vehicles and pedestrians. The amount and type of clutter is specified using clutter templates. Each template specifies a list of entity types with associated relative weights, as well as a

collection of control points. Each control point specifies a center location, a radius (defining a circular area), and a number of clutter entities. Each control point is identified as defining a static clutter area, a mobile clutter area, a clutter source, or a clutter sink. Clutter entities move randomly with a static clutter area. Sources and sinks allow dynamic traffic flows to be created. Clutter entities are randomly created in the source areas, and move to random locations within a sink area. When they arrive, they are destroyed and replaced with a new entity in one of the source areas. The clutter simulation publishes entity state information for each of the clutter entities as they move.

In support of the JFCOM Urban Resolve joint experiment, the clutter simulation has been significantly enhanced. A wide variety of clutter entities, including both vehicles and lifeforms, can be created. Templates can be defined that specify the movements of clutter entities at specific times. These templates can be used to more realistically model civilian traffic movements such as commuting to and from work, political demonstrations, etc.

2.2.5 MC02 Federation Object Model

The JSB-RD simulation environment currently uses a version of the Millenium Challenge 2002 (MC02) federation object model (FOM). The MC02 FOM is based on the Realtime Platform Reference (RPR) FOM, which was designed to map the original DIS protocols into an HLA environment. The MC02 FOM includes a number of objects and interactions that were added to support specific aspects of the Millenium Challenge 2002 exercise. Most of these are not used within the JSB-RD environment. The elements of the MC02 FOM that are currently used include:

- the BaseEntity/PhysicalEntity/Platform class, primarily the GroundVehicle and Aircraft subclasses,
- the Atmosphere, SurfaceWeather, and Weather classes, and their various subclasses,
- the Collision, Weapon Fire, and Munition Detonation interactions, and
- the GIEsim Entity State, Message Send, and Message Received interactions.

The JSB-RD environment generally conforms to the MC02 Federation Agreements, Spiral 3-10, May 2002.

2.2.6 Integrated Situation Viewer

The Integrated Situation Viewer, commonly referred to simply as “the Viewer”, is intended to support experimentation with theater-level air mission situation awareness and dynamic retasking. The Viewer displays simulation state information describing air missions, which is, in effect, ground truth, and the corresponding Air Battle Plan information. In addition, any information (derived from the simulation) that reflects the reported or perceived current situation, as output by various sensor models, is displayed. In its current form, the Viewer consists of a number of loosely coupled components. Figure 2.6 shows the architecture for the Viewer, which is based on the classic Model-View-Controller paradigm.

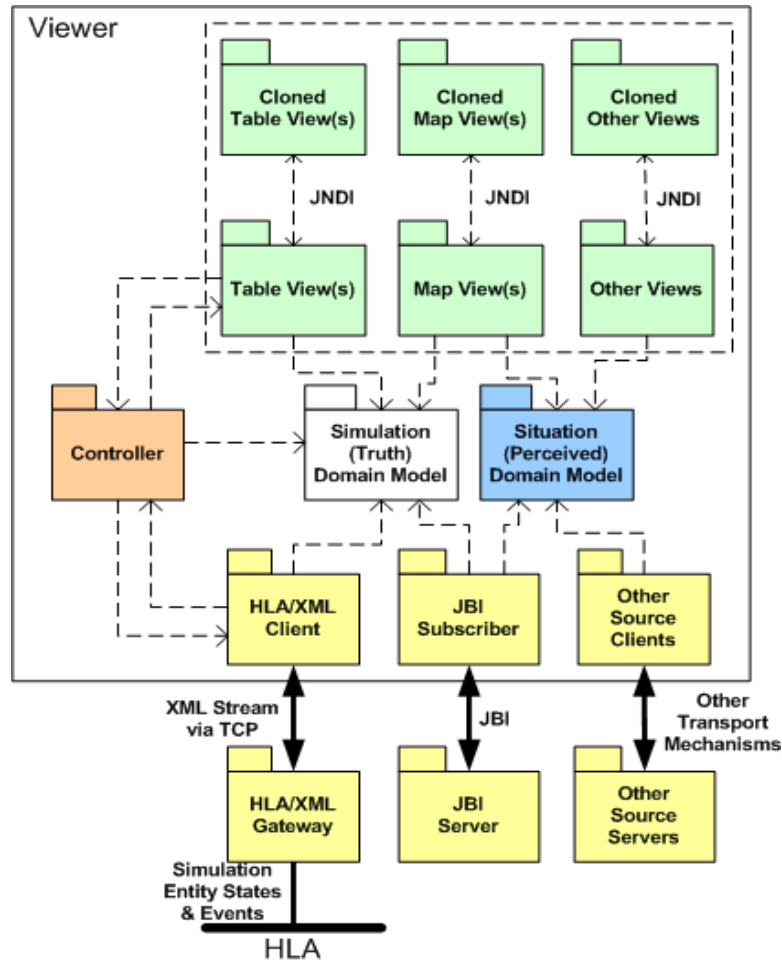


Figure 2.6: Viewer Architecture

The back-end portion of the Viewer consists of a collection of components that are capable of reading data from various sources. Simulation entity state and event data output by JSAF, OASES, DT-SIM, and other simulations is read via an HLA-to-XML gateway. This gateway converts entity state information received over HLA into a stream of XML messages. It also performs coordinate conversion (from the geocentric coordinates used by the MC02 FOM to geodetic coordinates), and partial translation of the DIS Entity Bit Vector (EBV) fields that are used to hierarchically identify entities by nationality, domain, type, subtype, etc. Other data sources, including JSAF input spreadsheets, are read directly from the appropriate files.

The “heart” of the Viewer consists of an integrated domain model that stores and maintains information on air mission plans, individual aircraft, and their targets. The data read from the various back-end sources is used to populate and update this model.

The front-end of the Viewer consists of multiple views of the information that the domain model contains. Several types of views are supported, including:

- Map views – displaying the locations of aircraft and targets, planned and actual flight paths, etc., overlaid on a map background at multiple scales and resolutions using the JView visualization toolkit developed by AFRL/IFSB (see Figure 2.7),
- Tabular views – listing entities of various types (aircraft, targets, missions, etc.) and their relevant characteristics, and capable of being sorted and ordered in various ways,
- Text views – displaying a streaming list of text messages, generated by the Simulation Network News (SNN) utility, which is part of the JSAF distribution.

The main Viewer GUI, shown in Figure 2.7, consists of a default Map View and a row of buttons at the bottom of the window to start other views. With the exception of the SNN View, multiple instances of all of the views can be launched.

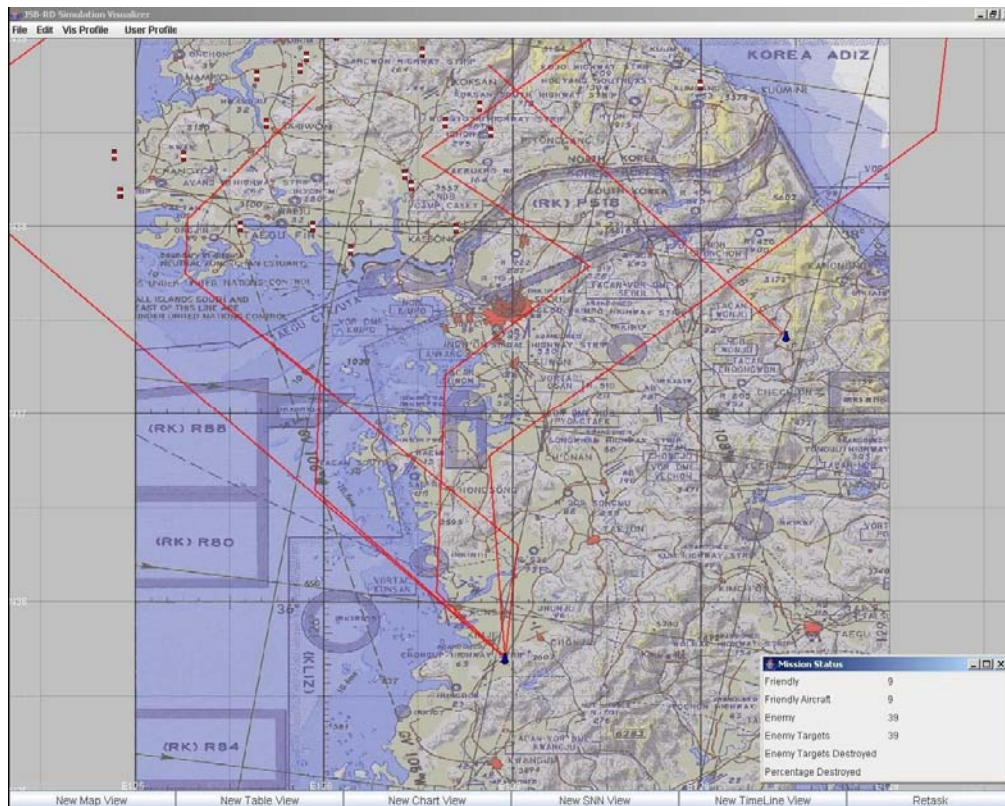


Figure 2.7: Main (Map View) GUI

The Map View contains entities, routes and background map data. It is also where the lens appears (see Figure 2.8).

The lens feature allows the user to left click anywhere in the main GUI, causing both a lens and a separate magnified Map View to appear. The lens can be moved by clicking and holding the mouse button within the title bar region of the lens window. Clicking in another location on the map will cause the lens to snap to that location. The movement of the lens is reflected by the map information in the Magnified Map View.



Figure 2.8: Lens View

The Magnified Map View, shown in Figure 2.9, shows the same data as the Map View but with both the CDRG and DTED map background at a higher resolution than the Map View. The center divider can be moved to adjust the amount of space both map types take up on the screen. Any entity icon selections made in the Magnified Map View will result in the selected item being highlighted in the Table View if it is visible there. Route selections made in the Magnified Map View will result in the Route Name being output to the command window.

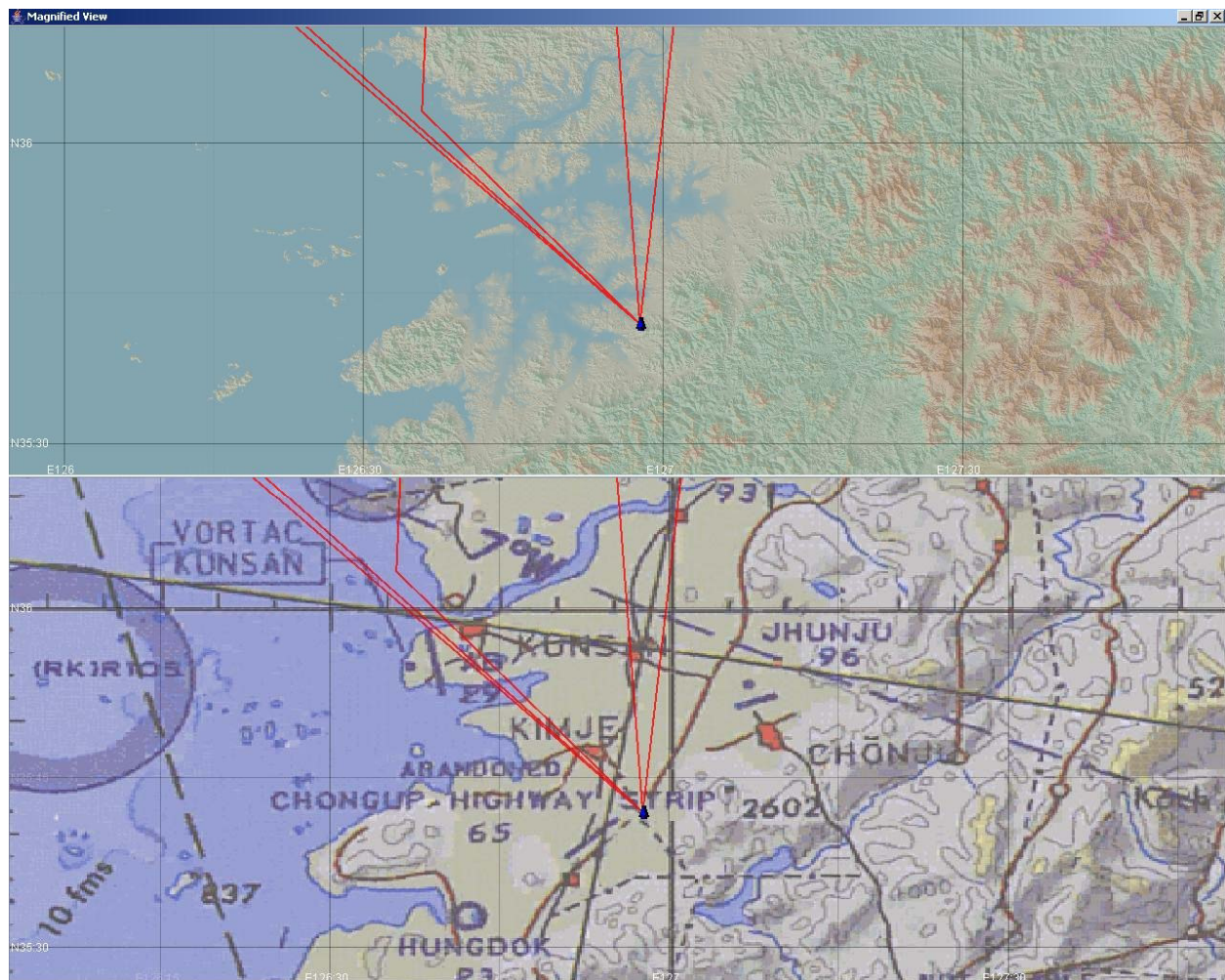


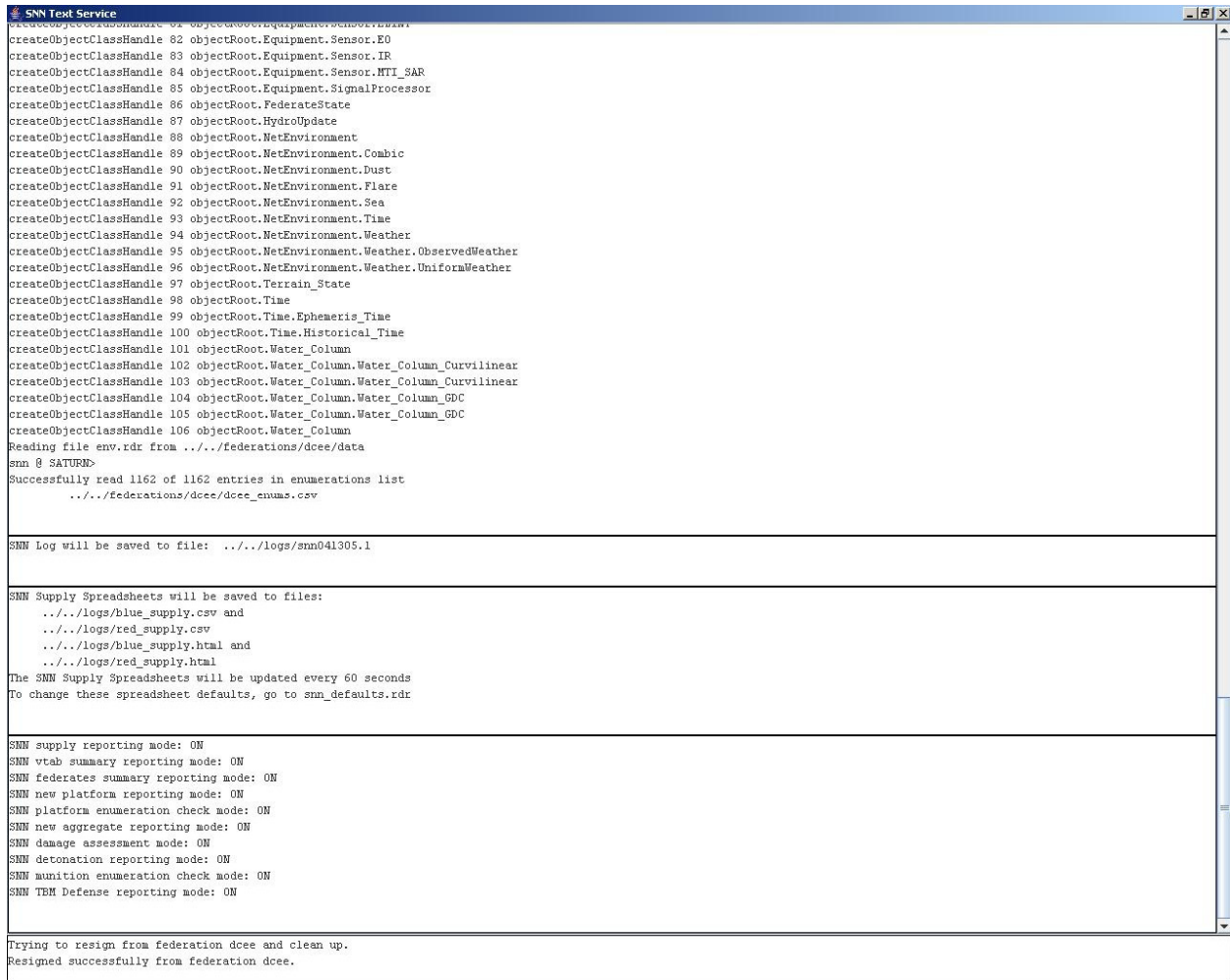
Figure 2.9: Magnified Map View

The Table View, shown in Figure 2.10, is launched via the New Table View button at the bottom of the main GUI window. This view contains data related to the entities shown on the map. The columns can be moved around and the rows can be sorted by clicking on the column header. Complex sorts are enabled by holding the control key down when clicking subsequent column headers. A row selected in the table causes the related entity icon to be highlighted on the Map View.

OID	Marking	E-Kind	MsnStatus	Damage State	Position	TargetLoc	Domain	Cntry	Cat	SubCat	Specific	Extra	ForceID
1	ALPHA01	1	0	0	35°41'54.0"N126°57'44.8"E		Air	225	2	11	1	0	Friendly
2	ALPHA012	1	0	0	35°41'54.0"N126°57'46.2"E		Air	225	2	11	1	0	Friendly
3	AC7	1	0	0	37°24'54.0"N128°27'10.7"E		Air	225	2	11	1	0	Friendly
4	DELTA01	1	0	0	37°24'54.0"N128°27'12.1"E		Air	225	2	11	1	0	Friendly
5	ALPHA03	1	0	0	35°41'54.0"N126°57'42.0"E		Air	225	2	11	1	0	Friendly
6	DELTA012	1	0	0	37°24'54.0"N128°27'13.6"E		Air	225	2	11	1	0	Friendly
7	BRAVO03	1	0	0	35°41'53.9"N126°57'51.7"E		Air	225	2	11	1	0	Friendly
8	ALPHA032	1	0	0	35°41'54.0"N126°57'43.4"E		Air	225	2	11	1	0	Friendly
9	BRAVO032	1	0	0	35°41'53.9"N126°57'53.1"E		Air	225	2	11	1	0	Friendly
10	E0000229	1	0	0	39°15'22.0"N127°17'41.0"E		Land	222	4	11	2	0	Opposing
11	E0000231	1	0	0	38°30'00"N126°53'16.4"E		Land	222	4	11	2	0	Opposing
12	E0000233	1	0	0	38°12'50.0"N126°28'02.9"E		Land	222	4	11	2	0	Opposing
13	E0000235	1	0	0	37°59'59.0"N125°56'34.3"E		Land	222	4	11	2	0	Opposing
14	E0000237	1	0	0	39°09'30.0"N126°02'20.8"E		Land	222	4	11	2	0	Opposing
15	E0000239	1	0	0	39°16'30.0"N126°36'47.3"E		Land	222	4	11	2	0	Opposing
16	E0000241	1	0	0	39°41'12.0"N127°22'19.8"E		Land	222	4	11	2	0	Opposing
17	E0000291	1	0	0	39°41'40.0"N125°45'10.3"E		Land	222	4	11	2	0	Opposing
18	E0000293	1	0	0	37°52'25.0"N126°08'41.5"E		Land	222	4	11	2	0	Opposing
19	E0000295	1	0	0	38°22'35.9"N124°53'25.0"E		Land	222	4	11	2	0	Opposing
20	E0000302	1	0	0	39°40'03.0"N124°40'56.7"E		Land	222	4	11	2	0	Opposing
21	E0000243	1	0	0	39°29'50.0"N125°42'27.1"E		Land	222	4	11	2	0	Opposing
22	E0000245	1	0	0	38°22'30.0"N125°15'32.3"E		Land	222	4	11	2	0	Opposing
23	E0000271	1	0	0	38°10'32.9"N124°55'08.7"E		Land	222	4	11	2	0	Opposing
24	E0000273	1	0	0	37°59'50.0"N125°33'30.1"E		Land	222	4	11	2	0	Opposing
25	E0000275	1	0	0	38°33'04.0"N126°38'41.7"E		Land	222	4	11	2	0	Opposing
26	E0000277	1	0	0	38°50'19.9"N125°52'35.2"E		Land	222	4	11	2	0	Opposing
27	E0000283	1	0	0	39°17'30.0"N127°29'54.8"E		Land	222	4	11	2	0	Opposing
28	E0000285	1	0	0	39°42'19.9"N127°40'26.4"E		Land	222	4	11	2	0	Opposing
29	E0000287	1	0	0	39°25'21.0"N125°30'38.8"E		Land	222	4	11	2	0	Opposing
30	E0000289	1	0	0	39°37'15.0"N125°20'49.3"E		Land	222	4	11	2	0	Opposing
31	E0000423	1	0	0	38°54'05.0"N125°54'32.9"E		Land	222	4	11	2	0	Opposing
32	E0000425	1	0	0	39°00'54.1"N126°04'11.3"E		Land	222	4	11	2	0	Opposing
33	E0000427	1	0	0	39°08'45.1"N127°22'48.5"E		Land	222	4	11	2	0	Opposing
34	E0000429	1	0	0	39°07'40.0"N127°30'05.0"E		Land	222	4	11	2	0	Opposing
35	E0000431	1	0	0	38°48'05.7"N125°38'55.4"E		Land	222	4	11	2	0	Opposing
36	E0000433	1	0	0	38°32'48.9"N125°41'51.6"E		Land	222	4	11	2	0	Opposing
37	E0000374	1	0	0	38°10'32.9"N124°55'28.8"E		Land	222	4	11	2	0	Opposing
38	E0000435	1	0	0	38°16'28.0"N126°26'02.2"E		Land	222	4	11	2	0	Opposing
39	E0000437	1	0	0	39°57'13.9"N124°39'23.8"E		Land	222	4	11	2	0	Opposing
40	E0000439	1	0	0	38°38'14.9"N126°49'44.4"E		Land	222	4	11	2	0	Opposing
41	E0000441	1	0	0	38°44'59.7"N127°59'47.7"E		Land	222	4	11	2	0	Opposing
42	E0000443	1	0	0	39°57'46.8"N127°18'43.7"E		Land	222	4	11	2	0	Opposing
43	E0000445	1	0	0	38°55'32.9"N126°31'30.5"E		Land	222	4	11	2	0	Opposing
44	E0000447	1	0	0	40°09'54.5"N126°43'48.7"E		Land	222	4	11	2	0	Opposing
45	E0000449	1	0	0	37°59'28.9"N126°42'06.2"E		Land	222	4	11	2	0	Opposing
46	E0000451	1	0	0	39°48'29.1"N125°11'43.0"E		Land	222	4	11	2	0	Opposing
47	E0000453	1	0	0	40°13'33.2"N127°26'47.8"E		Land	222	4	11	2	0	Opposing
48	E0000455	1	0	0	40°15'60.0"N127°39'34.3"E		Land	222	4	11	2	0	Opposing

Figure 2.10: Table View

The SNN View, shown in Figure 2.11, is launched via the New SNN View button at the bottom of the main GUI window. This view contains the text output by the SNN module found in the JSAF suite.



```

SNN Text Service
createObjectClassHandle 81 objectRoot.Equipment.Sensor.Buina
createObjectClassHandle 82 objectRoot.Equipment.Sensor.E0
createObjectClassHandle 83 objectRoot.Equipment.Sensor.IR
createObjectClassHandle 84 objectRoot.Equipment.Sensor.MTI_SAR
createObjectClassHandle 85 objectRoot.Equipment.SignalProcessor
createObjectClassHandle 86 objectRoot.FederateState
createObjectClassHandle 87 objectRoot.HydroUpdate
createObjectClassHandle 88 objectRoot.NetEnvironment
createObjectClassHandle 89 objectRoot.NetEnvironment.Combic
createObjectClassHandle 90 objectRoot.NetEnvironment.Dust
createObjectClassHandle 91 objectRoot.NetEnvironment.Flare
createObjectClassHandle 92 objectRoot.NetEnvironment.Sea
createObjectClassHandle 93 objectRoot.NetEnvironment.Time
createObjectClassHandle 94 objectRoot.NetEnvironment.Weather
createObjectClassHandle 95 objectRoot.NetEnvironment.Weather.ObservedWeather
createObjectClassHandle 96 objectRoot.NetEnvironment.Weather.UniformWeather
createObjectClassHandle 97 objectRoot.Terrain_State
createObjectClassHandle 98 objectRoot.Time
createObjectClassHandle 99 objectRoot.Time.Ephemeris_Time
createObjectClassHandle 100 objectRoot.Time.Historical_Time
createObjectClassHandle 101 objectRoot.Water_Column
createObjectClassHandle 102 objectRoot.Water_Column.Water_Column_Curvilinear
createObjectClassHandle 103 objectRoot.Water_Column.Water_Column_Curvilinear
createObjectClassHandle 104 objectRoot.Water_Column.Water_Column_GDC
createObjectClassHandle 105 objectRoot.Water_Column.Water_Column_GDC
createObjectClassHandle 106 objectRoot.Water_Column
Reading file env.rdr from ../../federations/dcee/data
snn 8 SATURN
Successfully read 1162 of 1162 entries in enumerations list
../../federations/dcee/dcee_enums.csv

SNN Log will be saved to file: ../../logs/snn041305.1

SNN Supply Spreadsheets will be saved to files:
../../logs/blue_supply.csv and
../../logs/red_supply.csv
../../logs/blue_supply.html and
../../logs/red_supply.html
The SNN Supply Spreadsheets will be updated every 60 seconds
To change these spreadsheet defaults, go to snn_defaults.rdr

SNN supply reporting mode: ON
SNN vtab summary reporting mode: ON
SNN federates summary reporting mode: ON
SNN new platform reporting mode: ON
SNN platform enumeration check mode: ON
SNN new aggregate reporting mode: ON
SNN damage assessment mode: ON
SNN detonation reporting mode: ON
SNN munition enumeration check mode: ON
SNN TEM Defense reporting mode: ON

Trying to resign from federation dcee and clean up.
Resigned successfully from federation dcee.

```

Figure 2.11: SNN View

The Controller is the “brain” of the Viewer, coordinating all of the other components. When new information arrives from one of the sources, the Controller triggers the updating of the domain model, and then the updating of all views that are affected by the change. Similarly, when the interactively changes a piece of information in one of the views, the Controller triggers the updating of the domain model, and then the updating of any other views that are affected by the change. The Controller also coordinates the selection of entities and locations across all of the views, so that an entity that is selected in one of the views is also highlighted in all other views in which it appears. A Controller GUI allows the user to select which types of views should be displayed and what the content of each should include.

A planned enhancement to the Viewer will make use of a custom implementation of the Java Naming and Directory Interface (JNDI), developed by AFRL/IFSB, to allow views to be “cloned” across multiple workstations and monitors.

2.2.7 Simulation Preparation

The Simulation Preparation component of the JSB-RD environment allows existing Air Battle Plans (ABPs) contained within the Air Operations Data Base (AODB) of the Theater Battle Management Core System (TBMCS) to be converted into sets of JSAF input spreadsheets that can be executed using JSAF. This component extracts a specified ABP from the AODB, which contains specifications of multiple air missions of various types. Each mission specification includes the numbers and types of aircraft involved in the mission, their takeoff and return times and bases, and a sequence of key mission events. These mission events include takeoff, refueling (start and end), time on target (start and end), and landing. Ground attack missions also identify their respective targets. Supporting information describing the mission targets is extracted from the MIDB.

The basic mission information, along with a list of the air defense threats in the area, can be fed to a separate Routing application, which determines the “best” route for each mission to and from its assigned target while avoiding air defense threats. The returned route contains a number of intermediate waypoints that the aircraft should pass through on the way to and from their target.

This information is then used to generate a JSAF input spreadsheet. The spreadsheet contains two entries for each scheduled air mission, one describing the ingressing leg of the mission, and the other describing the return leg. Each entry specifies, in JSAF terms, the number and type of aircraft, the call sign(s) of the aircraft, the type of task to be performed, the take off and return times, and the base and target locations. A second spreadsheet contains the intermediate route points, which are linked to the missions by name.

2.2.8 GIESim

The Global Information Enterprise Simulation (GIESim) provides high fidelity Link-16 network modeling with full resolution of propagation effects, including power and distance based Signal to Noise ratio, terrain masking and other Line Of Sight (LOS) issues. The vision of GIESim is to move, process, manage, and protect the C4ISR information that supports the functions of Global Awareness and Dynamic Planning and Execution. The mission of GIE is to link aerospace assets in-theater and globally, to integrate C3 & ISR networks, to defend critical information systems from cyber attack, and to develop new information processing and management techniques. Most large-scale force level simulations assume perfect communications. The lack of communications in a simulation environment can lead to the prediction of erroneous results. Tools are needed to bridge these communications modeling gaps.

The GIESim project vision is to define, design and implement a Modeling and Simulation (M&S) framework for the Global Information Enterprise (GIE). Within the GIESim framework users are able to execute, via a common interface, multiple communications and network M&S tools to effectively and efficiently analyze candidate communications architectures and technologies. The GIESim can interface with other M&S tools (e.g., force-level simulations and

detailed hardware system models) to provide the appropriate level of M&S fidelity and processing speed for the broad spectrum of M&S tasks. The GIESim user base spans advanced technology researchers to communications network architects to mission planners.

Within the JSB-RD federation, the role of GIESim is to evaluate communications connectivity between various entities in the simulation. Communication networks are defined within GIESim, tying together various collections of simulated entities. JSAF controls the movements of the simulated entities. GIESim monitors customized entity state messages published by JSAF and updates the locations and headings of the entities. When two entities need to communicate with each other, JSAF sends a request to GIESim identifying the two entities, the type of communication, and the length of the message. GIESim then determines whether or not the specified entities can communicate, either directly or via available relays. If they can communicate, GIESim returns a response to JSAF indicating the delay before the message will arrive at its destination. JSAF then schedules the delivery of the message at the indicated time.

3. LESSONS LEARNED

This section summarizes the lessons learned during this effort. The key accomplishments achieved under this effort include:

- enhanced capabilities to extend and modify JSAF,
- enhanced capability to use JSAF to execute Air Battle Plans extracted from the TBMCS Air Operations Data Base,
- enhanced HLA-to-XML gateway to translate distributed simulation message traffic over HLA into a stream of XML messages,
- enhancement of the Viewer to support multiple views of several different types.

Each of these discussed in the subsections below. Finally, conclusions and recommendations are given.

3.1 JSAF Modification

JSAF is not a packaged software product. JFCOM J9 maintains a staff of more than 30 developers who are constantly modifying and extending JSAF to meet the requirements of an ongoing series of simulation experiments. Although the JSAF development staff has been helpful in finding answers to questions, supporting external users of JSAF is not a high priority. Although there are a number of external users of JSAF, there does not appear to be an active JSAF user/developer community outside of JFCOM, with mailing lists, news groups, and other such resources, where newer users can go to find answers to their questions. There is no JSAF help desk, or any other similar support mechanisms.

With help from the JSAF developers, we have learned to make several types of minor enhancements to the JSAF software. These include:

- Adding a new type of platform/vehicle,
- Adding new components to an existing platform/vehicle,

- Adding a new type of military unit,
- Adding a new high-level taskframe for an air mission,
- Changing the values of parameters associated with a platform/vehicle or high-level task frame, and
- Adding new FOM interactions that can be sent or received by the JSAF federate.

The JSAF software contains an impressive amount of functionality. However, closer examination reveals that much of the functionality that appears to be present is difficult or impossible to access. JSAF has been modified extensively over the years to meet specific needs of particular exercises and experiments. It appears that many of these modifications were never completed, or never worked successfully. However, they remain in the JSAF software distribution. In some cases, it appears that capabilities were added at one point, but were later incompletely removed, leaving unused entries in data files, etc. For example, JSAF contains a number of software libraries that appear to support Link-16/TADIL-J communications functionality, as well as data elements. The MC02 FOM includes messages that describe Link-16 radios and TADIL-J messages. However, all attempts to make use of this functionality have been completely unsuccessful. The integration of JSAF and GIESim provides an alternative form of this functionality.

JSAF does a reasonable job of simulating the movement of physical entities from one location to another. It also does a reasonable job of simulating many types of individual weapons and the related combat activities. It can model tactical aircraft and simple air missions adequately. Although it does model some types of sensors, including the detection of targets, it does not model the tracking, reporting, or dissemination of information resulting from sensor detections to other entities. It does not model most ISR assets (e.g., AWACS, JSTARS, UAVs) with useful realism. It should be noted that for the current Urban Resolve experiment, JFCOM J9 is not using JSAF for sensor modeling, but instead relied on a COTS software package SLAMEMTM (Simulation of the Locations and Attack of Mobile Enemy Missiles) from Toyon Research Corporation.

The JSAF software distribution is extremely large and complex. It now consists of more than 1000 individual object libraries. A complex web of dependencies connects these libraries to one another. The JSAF software is primarily written in ANSI C, but has a complex object-based structure that simulates inheritance and aggregation relationships among libraries. Extensive configuration script and reader (i.e., data) files further add to this complexity. This complexity makes the JSAF software extremely difficult to understand. JSAF documentation is limited. Many of the libraries have no documentation whatsoever.

JSAF has two interfaces for creating scenarios. One is the interactive Unit Editor, which allows military units and platforms to be created, placed on the Plan View Display, and assigned tasks. Once created, scenarios can be saved to files for reloading later. The second interface is the Spreadsheet mechanism, which allows spreadsheets specifying air mission information to be read in, from which the necessary aircraft are created and then tasked. Spreadsheets can also be written out, but without any tasking information. Oddly, there is very little overlap between the aircraft tasks that can be assigned using these two mechanisms. Tasks that can be assigned interactively using the Unit Editor generally cannot be assigned using the Spreadsheet

mechanism, while the mission types that can be assigned using the Spreadsheet mechanism cannot be assigned using the Unit Editor. The Unit Editor also offers a number of interactive commands (e.g., Fly Higher/Fly Lower) that cannot be accessed through other mechanisms. In particular, there has been no mechanism that allows military units to be tasked via an HLA message from another federate. JFCOM controls JSAF-based scenarios using multiple operators, who interpret commands and interactively manipulate the JSAF entities accordingly. Because of this, JFCOM sees little need for such a capability. However, in order to interface JSAF with real C4ISR systems, this capability is critical. The modifications made to JSAF to integrate it with GIESim provide the beginnings of such a capability, allowing JSAF to respond to either text messages or HLA interactions.

3.2 JSAF Scenario Preparation

The ability to extract an Air Battle Plan from the TBMCS Air Operations Data Base, along with supporting target information from the MIDB, and to generate a JSAF input spreadsheet that can be used to execute those air missions, is a very powerful tool. When used in conjunction with a realistic mission route generator that takes into account known air defense threats, it provides an even more important capability.

However, there remain several limitations to this capability that should be addressed. Currently, multiple queries are performed directly on AODB and MIDB tables. The information acquired is then manipulated to create a JSAF input spreadsheet. It would be more convenient if all of the necessary information could be easily queried, via JBI, IOTA, or another similar mechanism, by specifying an Air Battle Plan name. It would also be convenient if a list of the Air Battle Plans contained within the AODB was easily accessible. Version 1.1.3 of TBMCS provides a collection of web services which may facilitate access to this information.

There are also several issues concerning the translation of ABP information into JSAF input. AODB aircraft types must be translated into corresponding JSAF entity types. The necessary JSAF entities are created for each specified air mission. This makes it difficult for a particular aircraft to fly multiple missions in sequence. ABP mission numbers are not used by JSAF. Instead, the call signs for each mission in the ABP are augmented with unique numeric IDs, and are inserted into the “markings” field of the JSAF entities. These call signs are not always unique. The mission types specified in the ABP must be translated into JSAF air mission task frames. Currently, the task frames used are Interdiction/Attack Ground for strike missions, FWA Hold for orbiting (including CAP, tankers, etc.), and Return to Base. Finally, the route points that can be entered into JSAF are two-dimensional (latitude-longitude). Altitude is specified indirectly, as a parameter of the top-level task frame. This makes it difficult to specify mission profiles that depend on variations in altitude.

3.3 HLA-to-XML Gateway

The HLA-to-XML gateway allows multiple applications, such as the Viewer, to access the stream of information coming out of the simulations without having to deal with the complexities of HLA. However, the HLA-to-XML gateway currently has several limitations. It currently provides only entity state information. It does not currently publish MC02 FOM interactions, such as collisions, weapon fire, and munition detonation events. The entity state

information is published as a simple sequence of entity states. However, as a result of using RTI-S, these are not time stamped; they are simply provided “in real-time” as they are received. Also, the translation of entity state fields is currently limited to coordinates, force identifier, and damage state, leaving the application to deal with the more difficult translation of the seven EBV fields that identify the type of the entity. Finally, the HLA-to-XML gateway currently operates in only one direction. Therefore, it only supports “receive-only” federates. Any HLA messages that an application needs to send back to the simulation to, for example, retask an aircraft, must use another mechanism.

3.4 Integrated Situation Viewer

The Integrated Situation Viewer has several limitations. Some of these are based on the data sources that it uses. The Viewer does not currently use Air Battle Plan or MIDB target information directly. Instead, it uses the JSAF input spreadsheet as its sole source of air mission planning information. As a result, it does not see all of the information in the Air Battle Plan, but only the subset that JSAF can make use of. The limitations of the HLA-to-XML Gateway described above also constrain the Viewer. The Viewer does not currently receive interaction messages, and the entity state messages that it receives have no time stamps. This makes it impossible for the Viewer to compare the planned mission timing with the timing of the simulated mission execution. Finally, while the Viewer displays information in various forms, much more could be done to filter, summarize, and aggregate the information to better meet the needs of a JFACC or subordinate commander.

3.5 JSAF-GIEsim Integration

The JSAF-GIEsim integration provides an excellent starting point for creating enhanced capabilities. By enabling JSAF to receive and process communications, it provides a basic mechanism that allows air missions in JSAF to be tasked or retasked by another federate, such as the Viewer. It also provides a mechanism that allows aircraft in JSAF, as well as other entities, to communicate with one another in a realistic manner. Further, the implementation of this capability has been a vehicle for learning a great deal about the internal operation of JSAF. However, a key limitation of this capability is the need to design the Link-16 networks that are to be used by JSAF entities. Also, JSAF entities and GIEsim entities must be created independently, and their identifiers must be mapped to each other.

3.6 Conclusions and Recommendations

The conclusions and recommendations resulting from this effort are summarized below:

1. JSAF has been used successfully by a number of large programs, including Joint Strike Fighter (JSF), EnviroFed, and Distributed Mission Training (DMT), and the use of JSAF facilitates the development of a closer relationship between AFRL and JFCOM. However, JSAF is not very suitable for use within a small, dynamic laboratory environment. It requires a large, highly trained, and highly skilled staff to maintain, to operate, and especially to modify or extend it. Extending or modifying the JSAF software in support of a specific simulation experiment is difficult and time-consuming. Although it can be integrated with other existing tools and C4ISR systems, this is not easy to accomplish. Continued use of

JSAF within the AFRL/IFSB modeling and simulation laboratory will require a significant commitment to develop and maintain a skilled and experienced staff.

- a) Obtain and install the current version of JSAF on a regular (semi-annual or annual?) basis. This is necessary in order to remain reasonably in synchronization with the JSAF development staff at JFCOM.
 - b) Continue to investigate the JSAF software and to learn how to modify and extend it. This includes:
 - i) Extending the spreadsheet input mechanism to support additional types of air missions.
 - ii) Learning to create new air mission task frames, and to modify existing air mission task frames.
 - iii) Learning how to extend the initial capability to support the tasking and retasking of entities from external applications, such as the Viewer.
 - c) Consider augmenting the use of JSAF with other, more modern entity level simulations that are focused on air operations, such as JIMM and/or EADSIM, and/or higher-level Air Force simulations such as AWSIM. Copies of these simulations should be obtained, installed, and evaluated with respect to their usability and flexibility.
2. Extend the Scenario Preparation software to publish the Air Battle Plan and associated order of battle information in XML form, unless this information can be obtained directly in XML form using TBMCS version 1.1.3. This information can then be input to the Viewer, allowing the planned missions to be more effectively compared with the actual execution of those missions.
 3. Extract Air Control Measure information from the AODB. This information should be used by the Router to create more realistic routes, and should be displayable by the Viewer.
 4. Improve the XML-to-HLA gateway to support MC02 FOM interactions including weapon fire and munition detonation, takeoff and landing, and entity removal, and to add a time stamp to the XML representation of each published HLA message. This can be accomplished by recording the wall-clock time at which the XML-to-HLA gateway joins the federation, noting the wall-clock time when each HLA message is received, and using the difference as the time stamp for that message.
 5. Publish simulation events that correspond to planned mission events, including takeoff and landing, weapon fire and munition detonation, start and end of refueling, etc., via the HLA-to-JBI gateway, so that they may be accessed by CAOC-21 applications.
 6. Enhance the Viewer to display additional air mission-related information, and to display information in additional forms. This includes:
 - a) Add support for Java Naming and Directory Interface (JNDI) technology, to allow all types of views to be duplicated across multiple monitors on multiple systems, while remaining synchronized with one another.

- b) Add support for retasking air missions. Given a specified target, determine which air missions can be retasked to attack that target, based on their location, status, weapons load, fuel, and other factors.
 - c) Display of MC02 FOM interactions, including weapon fire and munition detonation, and takeoff and landing, as well as entity removal.
 - d) Display of planned mission information from the Air Battle Plan, and graphical comparison of planned mission states with actual mission states.
 - e) Time line/Gantt chart type displays showing planned and actual mission event (takeoff, on target, return, etc.) times. This has been begun, using software from the ATAOC effort, but has not yet been completed.
 - f) Additional graph and chart views which show aggregate mission success, in terms of aircraft lost and/or targets destroyed or disrupted.
 - g) Display of Air Control Measures, which show restricted airspaces.
 - h) Display of weather conditions, in 2D or 3D, as published by OASES.
 - i) Three-dimensional views that can be attached to a specific aircraft or group of aircraft.
 - j) Vector map data (e.g., VMap Level 0, 1, or 2) overlaid on terrain elevation data.
 - k) Addition of improved controls for customizing individual views, as well as creating, naming, storing, and retrieving collections of views including view configuration (i.e., window size and placement) information.
7. Enhance the JSAF-GIEsim integration to provide a more robust and flexible capability. This includes:
- a) To the maximum extent possible, enhance the Scenario Preparation software to automate the generation of all necessary GIEsim scenario input files, including network design and entity identifier mapping. This will facilitate the development of new scenarios.
 - b) Consider adding an additional component to selected JSAF entities (i.e., aircraft) to allow them to send and receive various types of Link-16 messages in response to simulation events (e.g., takeoff, attack, landing) or time intervals (e.g., PPLIs).

4. ACRONYMS

This section provides a listing of acronyms and their meanings as used in this document.

ABP	Air Battle Plan
AFRL	Air Force Research Laboratory
AOC	Air Operations Center
AODB	Air Operations Data Base
ATO	Air Tasking Order
ATAOC	Advanced Technology Air Operations Center
AWACS	Airborne Warning and Control System
C3	Command, Control, and Communications
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CADRG	Compressed ARC Digital Raster Graphics
CDRL	Contract Data Requirements List
CGF	Computer Generated Forces
COA	Course of Action
COTS	Commercial Off The Shelf
CPE	Commander's Predictive Environment
CRC	Control and Reporting Center
CSE	Common Simulation Environment
DARPA	Defense Advanced Research Projects Agency
DIS	Distributed Interactive Simulation
DMSO	Defense Modeling and Simulation Office
DMT	Distributed Mission Training
DoD	Department of Defense
DSAP	Dynamic Situation Awareness and Prediction
DTED	Digital Terrain Elevation Data
DT-SIM	Dynamic Terrain Simulation
EBO	Effects Based Operations
EBV	Entity Bit Vector
EOB	Enemy Order of Battle
ESC	Electronic Systems Command
FOM	Federation Object Model
FWAEE	Fixed Wing Aircraft Exercise Editor
GIE	Global Information Enterprise
GIEsim	Global Information Enterprise Simulation
GUI	Graphical User Interface

HE	Human Effectiveness Directorate (of AFRL)
HLA	High Level Architecture
IFSB	C4ISR Modeling & Simulation Branch
IF	Information Directorate (of AFRL)
IFT	Technology Division
ISR	Intelligence, Surveillance, and Reconnaissance
JB	Joint Battlespace Infosphere
JEFX	Joint Expeditionary Force exercise
JFCOM	Joint Forces Command
JIMM	Joint Integrated Mission Model
JIPTL	Joint Integrated Prioritized Target List
JNDI	Java Naming and Directory Interface
JSAF	Joint Semi-Automated Forces
JSB	Joint Synthetic Battlespace
JSB-DS	Joint Synthetic Battlespace – Decision Support
JSB-RD	Joint Synthetic Battlespace for Research and Development
JSIMS	Joint Simulation System
JSTARS	Joint Surveillance and Target Attack Radar System
LOS	Line of Sight
MAAP	Master Air Attack Plan
MARCI	Multi-system Automated Remote Control and Instrumentation
MC02	Millenium Challenge 2002
METOC	Meteorological/Oceanographic
MIDB	Modern Integrated Data Base
M&S	Modeling and Simulation
ModSAF	Modular Semi Automated Forces
NGMS	Northrop Grumman Mission Systems
NIMA	National Imagery and Mapping Agency
OASES	Ocean, Atmosphere, and Space Environmental Services
OPFOR	Opposing Force
PBA	Predictive Battlespace Awareness
PFM	Pressure Field Modification
PSM	Portable Space Model
PVD	Plan View Display
RPR	Realtime Platform Reference
RRS	Rome Research Site
RTI	Run Time Infrastructure
SAA	Situation Awareness and Analysis
SAM	Surface to Air Missile

SEDRIS	Synthetic Environment Data Representation and Interchange Specification
SNE	Synthetic Natural Environment
SNN	Simulation Network News
STF	SEDRIS Transmittal Format
STOW	Synthetic Theater of War
SWEG	Simulated Warfare Environment Generator
TAOS	Total Atmosphere Ocean Services
TBMCS	Theater Battle Management Core System
UAV	Unmanned Aerial Vehicle
USATEC	U.S. Army Topographic Engineering Center
USSPACECOM	U.S. Space Command
VMap	Vector Smart Map
VPF	Vector Product Format
XML	eXtensible Markup Language